



Technical Report
CMU/SEI-95-TR-018
ESC-TR-95-018

Carnegie-Mellon University
Software Engineering Institute

**Fingertip Access to Software Engineering Information and Learning: SAIL on
the Informedia DVLS**

Harvey K. Hallman

May 1996

DISTRIBUTION STATEMENT 5

Approved for public release
Distribution Unlimited

DTIC QUALITY INSPECTED 3

Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of, "Don't ask, don't tell, don't pursue," excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

Technical Report

CMU/SEI-95-TR-018

ESC-TR-95-018

May 1996

Fingertip Access to Software Engineering Information and Learning: SAIL on the Informedia DVLS



Harvey K. Hallman

Software Engineering Information Modelling

19960723 022

Unlimited distribution subject to the copyright.

Software Engineering Institute

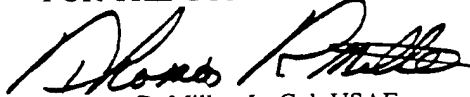
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER



Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

Copyright © 1996 by Carnegie Mellon University.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

Requests for permission to reproduce this document or to prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212.
Phone: 1-800-685-6510. FAX: (412) 321-2994. RAI also maintains a World Wide Web home page. The URL is <http://www.rai.com>

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145. Phone: (703) 274-7633.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Table of Contents

1	Introduction	1
2	Fingertip Access is Required	3
2.1	The Amount of Information	4
2.2	The Cost of Re-invention	4
2.3	Types of Information	6
2.4	Access to Information	6
2.5	Access To Training	7
2.6	Summary	7
3	Fingertip Access with SAIL on Informedia DVLS	9
3.1	Be Available at the Software Engineer's Fingertips	9
3.2	Support of Multimedia Information	10
3.3	Easy Access to Find the Desired Information	12
3.4	Produce a High Find/Seek Hit Ratio	12
3.5	Be Able to Expand With Project Information and Knowledge	13
3.6	Be Adaptable to Current Development Activities	14
3.7	Be Organized Into Sub-libraries	14
3.8	Provide for Both Learning and Reference	14
3.9	Shortcomings	15
4	Fingertip Access with the World Wide Web	17
4.1	The Web and Software Engineering	17
4.2	Accessing Information on the Web	17
4.3	Experience Makes it Better	18
4.4	Shortcomings	18
5	Fingertip Access with SAIL and the Web	19
5.1	At the Beginning of the Project	19
5.2	During the Project	19
5.3	On a Daily Basis	19
5.4	Fingertip Access in Large Organizations	20
6	The SAIL Approach to Using Informedia	21
6.1	Access and Learning	21
6.2	Library Organization	25
6.3	Facilitating Easy Access	28

7	Summary	29
	Appendix A SE-SAIL Library Outline: An Example	31
	Appendix B Content of a Major Topic	37
	References	42

List of Figures

Figure 1:	Example of a Find Selection Window	11
Figure 2:	Access Hit Ratio	13
Figure 3:	Accessing More Information About a Reference	23
Figure 4:	Library of Software Engineering Window	24
Figure 5:	Parts of a Major Topic	25
Figure 6:	Logical Hierarchy of the Repository	27
Figure B-1:	Example of the Master Reference List	40
Figure B-2:	Example of the Master Terminology List	41

Fingertip Access to Software Engineering Information and Learning: SAIL on the Informedia DVLS

Abstract: Practicing software engineers have difficulty accessing state-of-the-practice technologies in a timely manner. As a result, software engineers frequently re-invent the technology. Fingertip access to large amounts of information (project, state-of-the-practice, state-of-the-art, domain specific, etc.) should be provided so that software engineers perform their profession more effectively. The System for Access to Information and Learning (SAIL) approach on the Informedia™ Digital Video Library System (DVLS) demonstrates the feasibility of providing fingertip access to information and learning materials, using requirements elicitation as the technology base. Also, Informedia DVLS and the World Wide Web can supplement each other; by using the Web to gain access to certain areas of software engineering, we can begin to assemble libraries of technical information. Informedia DVLS can provide timely access to such information and can also be used to house current project materials.

1 Introduction

The SAIL effort, undertaken in the Software Engineering Information Modeling (SEIM) Project at the Software Engineering Institute (SEI), examined the feasibility of creating a system that, using the Informedia™ Digital Video Library System [Christel 95] as the vehicle for storage access and manipulation of multimedia information, could provide software engineers with access to the technical information that they need on their desktops. This system was developed jointly by SEI and the Computer Science Department at Carnegie Mellon University.

SAIL addresses the structural organization of and access to the technical information of a repository that contains information of different types of media. SAIL was an experiment at the SEI constructed to solve the problem of the common use of information for both reference and learning, to resolve the organizational questions associated with very large technical repositories, and to define approaches for placing technical information from many diverse sources onto a technical repository that allowed for easy access. The technical report CMU/SEI-95-TR-006 [Hallman 96] defines four strategies and nine approaches to preparing and recording both formal and informal technical information into a technical repository so that a software engineer using a support system like Informedia DVLS can access it. This technical report discusses the means to access the information on the repository and the approaches that will make that information useful for both reference and training.

SAIL on Informedia DVLS attempts to provide fingertip access (also known as FTA) and learning to software engineers. The World Wide Web [Berners-Lee94] (herein referred to as the Web) provides a limited amount of support for fingertip access, but by using the Web to find relevant documents and making then making these documents accessible with SAIL, fingertip access can become a reality. SAIL provides a structure for the software engineer to have fingertip access to reference and training materials not only from the state-of the-practice and state-of-the-art repositories but also to specific project information so vital to projects' success.

We must consider what a SAIL Library might look like if it contained all of the materials that the software engineer needs to do his or her job as a professional. This is referred to as SE-SAIL, or a "SAIL Library for Software Engineering."

Although this paper is about improving the productivity of software engineers and the quality of their products, its content applies to many other information-intensive professions.

2 Fingertip Access is Required

Fingertip Access (FTA) implies that the individual has access to the repositories of information by simply requesting it at his or her work station. The repositories can be accessed without leaving the work that is active on the display. The information found can be assimilated into the work being done. The software engineer not only needs fingertip access to project information but also needs access to reference materials containing the state-of-the-practice and occasionally the state-of-the-art plus training materials on how to use the reference materials. To make matters more demanding, the information needed is no longer just textual. It is often graphic or audio or video. The absence of fingertip access has resulted in massive re-invention within the project, the company and the industry.

The amount of information required to be available to the software engineer has been an historic problem. Not only has it become voluminous, but it is getting harder and harder to find when it is required. In the 1950s, technical information and software development results were maintained by the individual programmers in their office on shelves, in files, and in their desk. In the 1960s, as larger systems began to be developed using teams of programmers, special rooms were created to contain and manage the information produced by a software development project. In the 1970s, most of this information was moved onto massive disk drives with access by large computers. Some companies, like IBM, used entire buildings to house the electronic versions of their active software development projects. In the 1980s, the personal computer created the demand for access to these software development files from the desk-top.

Through these emerging years, access to the state-of-the-practice and state-of-the-art outside of the project has been through the use of libraries of physical documents. The content of these physical libraries are organized by publication and title rather than being organized for quick access to the technical content across the publications. In addition, these libraries were usually remote from the software development people. As a result, they have not been accessed effectively and re-invention has been the norm rather than the exception.

In the 1990s, the emergence of audio/video and CDROM technologies on personal computers has for the first time allowed the capture and then access to massive amounts of information at the software developers desk-top. The technologies now exist that can provide access not only to software development work products, but also to the project technical information as it is formulated, and to the state-of-the-practice and art from electronic libraries that can now be searched by content rather than by title. The user can also use the work station to learn how to use the technology.

Fingertip access is not only required but is now feasible.

2.1 The Amount of Information

In 1989, a study was conducted by the author at the SEI to determine the approximate maximum size of a fully populated "Software Engineering Handbook." The study addressed the size of a handbook printed in reference format on paper avoiding duplicate information but containing most of the appropriate information that is needed by a software engineer. The study concluded that if limited to state-of-the-practice only, it would be greater than 30,000 pages. This is about ten volumes if printed on tissue paper.¹ Add to this the state-of-the-art and local project information. The volume will be at least four or five times as great. In addition, as is true with many technical disciplines, the amount of information needed by the software engineer to be effective is increasing every year. Without access to it and/or knowledge of its existence, the software developer will continue to re-invent the needed technologies.

As part of the effort in understanding the feasibility of SAIL, an attempt was made to define the maximum outline for "Software Requirements Engineering." Outlines from seven sources were merged eliminating duplicates. These sources were texts, courses, and lectures on the subject matter. The result was twenty four pages of outline. This was then expanded into an outline of all of the potential knowledge needed and then condensed for this paper. Appendix A on page 31 shows one possible organization. It is helpful in understanding why this library is potentially so large. For example, the twenty four pages for the Category "Software Requirements Engineering" is summarized into eight lines in this outline.

2.2 The Cost of Re-invention

Re-invention has become acceptable, but it is costly.

The software engineer spends a lot of time either looking for technical information or re-inventing it. Unlike other engineering professions, the software engineer practicing in industry is more likely to re-invent a technical algorithm than to search for the state-of-the-practice and then build on it. Access to the appropriate information at the time it is needed is difficult and time consuming. Therefore, it is often ignored. This re-invention is costly and is one of the main reasons why good software quality is so evasive.

Software engineers have had access to technical libraries for years. These have been growing and becoming more complete in their information content. But in projects developing real software that is to be developed on schedule and with fixed resources, the technical libraries are seldom used. There are a number of reasons for this:

¹. *Perry's Chemical Engineers' Handbook* [Perry 84], is about 3,000 pages printed on very thin paper.

- Interruptions are costly in software engineer time. Physical technical libraries are usually not readily accessible. Even when management has made special efforts to have the libraries located near the project development personnel, the software engineer has to interrupt the work he or she is doing to go and find out whether the needed technical resource is available. Too frequently the needed resource is there but cannot be found. It may be buried inside an article in a technical journal. The book may be out on loan.
- Time is critical, so the software engineer often will delay the research of the library until a later time, which may never come.
- It is often easier to re-invent than it is to find existing technology.
- Training for the technology is often unavailable when it is needed and can cause long interruptions that need to be scheduled and often approved by management.

It has become acceptable to ignore technical libraries.

Re-invention increases the cost of the system by increasing the amount of testing, validation and verification necessary to assure the algorithm. It usually increases the length of time necessary to produce the re-invented algorithm. It also increases the risk the user takes with the unproven routine, adding to the probability that the routine will not work correctly or handle all of the necessary cases.

Project information is evasive and hard to locate. This information is usually on paper tucked away in the hidden corners of the project. Finding it is based on memory or by getting another copy of the document. Complete descriptions are usually unavailable when needed. Some of the reasons project information is ignored are the following:

- It is easier to ask someone how to do something than to dig it out for oneself. If the developer can find someone to tell him or her what is needed, their description will be used instead of the more formal documentation or presentation. It saves time but loses the why and the how discussed in the presentation of the technology needs.
- Techniques developed on previous projects are not easily accessible to current and future projects. There usually isn't an easy way to share such information. The transfer of the technology to the new project is often done by transferring the people with the knowledge to the new project.
- Much of the information is in multimedia form and therefore not available to the software engineer (e.g., presentations). The information may have been given in a presentation before the software engineer came onto the project. Or the software engineer didn't know he or she was going to be needing the information and didn't have time to attend the presentation.

For these and many other reasons, it is easier for the software engineer to re-invent a technology than to build on technologies that already exist.

2.3 Types of Information

The repository needs to contain a number of types of technical information in order for it to be useful to the software engineer. It needs to contain the following types:

- software engineering techniques - process, development, analysis, implementation
- computer science techniques - languages, algorithms, data, architectures, operating systems, storage management, etc.
- software examples (often referred to as reuse) - designs, code, test cases
- domain specific techniques - architectures, common algorithms, standards, etc.
- local techniques - process, studies, design, implementation, post deployment
- project specific information - external approved, project approved, intermediate results documents, presentations, decision meetings etc.
- hardware descriptions - computer and related hardware architectures, hardware specifications
- mathematics related to software engineering - set theory, probability, statistics, discrete math, algebra, formal methods, etc.

Software engineers are required to address new domains of knowledge every three to five years. This varies by individual and organization. But, as one project nears completion new areas for the use of computers are defined and the experienced software engineer is expected to help address the new area. It can take six months to a year for a software engineer to become proficient in the new domain, partially because access to the latest relevant information about the domain is not available in an easily accessible manner.

2.4 Access to Information

Information required by the software engineer in the 90s and beyond contains text, graphics, audio, and video. Multimedia is needed not only for technical state-of-the-practice and state-of-the-art but also for project information. Technical presentations frequently contain more information than just text. Equal access by the software engineer is required for all media forms. For example, the video and audio forms as well as text forms need to be searched on content. Graphics need to be able to be found based on content. Title information searches are often inadequate for these searches.

The software engineer needs to be given assistance in finding information; it cannot be assumed that all software engineer will refer to a technology with the same terminology. The search mechanisms should be based on a dialogue with the user, and number of aids need to be provided, such as synonym suggestions. Some of these methods such as guides and advisors are discussed in [Wood 94].

The presentation of the information that has been found needs to be positioned to the location of the requested information. Since the technical reference needed is often in the middle of a lengthy discussion of a broader topic, simply referencing the document or video leaves the user with the time-consuming problem of locating the reference within the larger entity. The media needs to be positioned at the spot where the first match was found.

If assisted by intelligent searching mechanisms, access to the information will be quick and easy.

2.5 Access To Training

Training and refresher courses on technical subjects (such as how to use an algorithm or a tool) interrupt the creative work activities. It may require special scheduling to be at some other place in the future. This extra effort often causes people to decided to delay using the technology, to use it and hope for the best, or to forego using it altogether. Delays in understanding an algorithm will frequently cause software engineers to use an alternate design.

There are few reasons why effective training cannot be accomplished at the work station in the form of fingertip learning. SAIL developed a prototype of a system for fingertip learning using requirements elicitation as the technical subject.

2.6 Summary

In order for the software engineer to perform his or her required tasks efficiently, with high quality results and at a predictable cost, the approach to accessing technologies has to change. Fingertip access to large amounts of information (project, state-of-the-practice or state-of-the-art, domain specific, etc.) will allow the software engineer to be more effective in performing his or her profession. It can also provide the software engineer with immediate training when the training is needed.

3 Fingertip Access with SAIL on Informedia DVLS

The work on the SEIM project demonstrated that this need can be solved. It can be solved by the software engineer having the technical information available at his or her work space to be used without a major interruption. The technical information at the state-of-the-practice can be built upon. Information at the state-of-the-art is helpful in guiding the new development work necessary for the project.

The SAIL approach provides an Informedia DVLS repository of technical information. (See Section 6 on page 21 below.) With SAIL, software engineers can have a sufficiently complete repository at their fingertips so that they can find the relevant technologies (both textual and multimedia) needed for their projects. The support system needs to

- be available at the software engineer's fingertips
- support multimedia information
- provide easy access to the desired information
- produce a high find/seek hit ratio
- be adaptable to current development activities
- be able to expand with project information and knowledge
- be organized into sub-libraries
- provide for both learning and reference

SAIL addresses the structure, organization and access to the technical information in a repository of technical information that contains information of many media types. (See Section 6 on page 21 below.) SAIL was an experiment at the Software Engineering Institute aimed at solving the use of information for reference and learning, resolving the organizational questions associated with very large technical repositories, and defining approaches to placing technical information from many diverse sources onto a technical repository that allows for easy access.

3.1 Be Available at the Software Engineer's Fingertips

Software engineers need certain information to be available on their desktop. A typical software engineer or software developer doesn't have time to go to a physical library; their design ideas can be lost or forgotten unless they have access to the information immediately. The software engineer must also be able to bring the technology found into his or her operational environment on the desktop. A library shelf in the work environment provides some access to information but doesn't enable the engineer to incorporate this information into the operational system he or she is creating.

Not only does the reference information need to be available but also the information needed to learn how to use the technology. This is referred to as "fingertip learning." The training information may be as simple as several paragraphs describing the referenced technology or as complex as a tutorial on its use. With SAIL on the Informedia DVLS, engineers can search for the desired algorithm electronically and then proceed to view the information needed in order to understand its logic and how to use it. They should also have the option to view general or detailed amount of information, depending on their needs. Ideally all searching and learning could be completed while the work in progress is waiting in another window; the newly discovered logic can then be picked up and adapted to the work in progress. This would eliminate the need to make trips to another room or another site and to copy or scan the information that would later need to be typed into the computer.

3.2 Support of Multimedia Information

Much of the information that software engineers need is not documented in text form. Considerable amounts of technical information, especially project information, are contained in discussions and presentations that are or can easily be recorded in audio or video format. Technical meetings within a project often lack sufficient documentation for those who cannot attend. Recording the presentations and discussions will keep this information from being lost.

With SAIL on the Informedia DVLS, software engineers or an administrative people can place recorded technical discussions, presentations, and decision meetings in text, audio video, or graphic form into the repository. Such presentations will then show when the user searches which matches their content, not just when a user searches for that media type. If an engineer wishes to find all of the references to "elicitation of functional requirements," a display will present all found incidents whether they be in text form or on soundtracks of audios or videos (see Figure 1). By pressing on the associated buttons, the file is brought up onto the screen, positioned at the location where the information has been found. If the source is text, the text that matches the request or synonyms to the request will be highlighted. If the source is audio or video, it will be positioned just in front of the place where the request was matched.

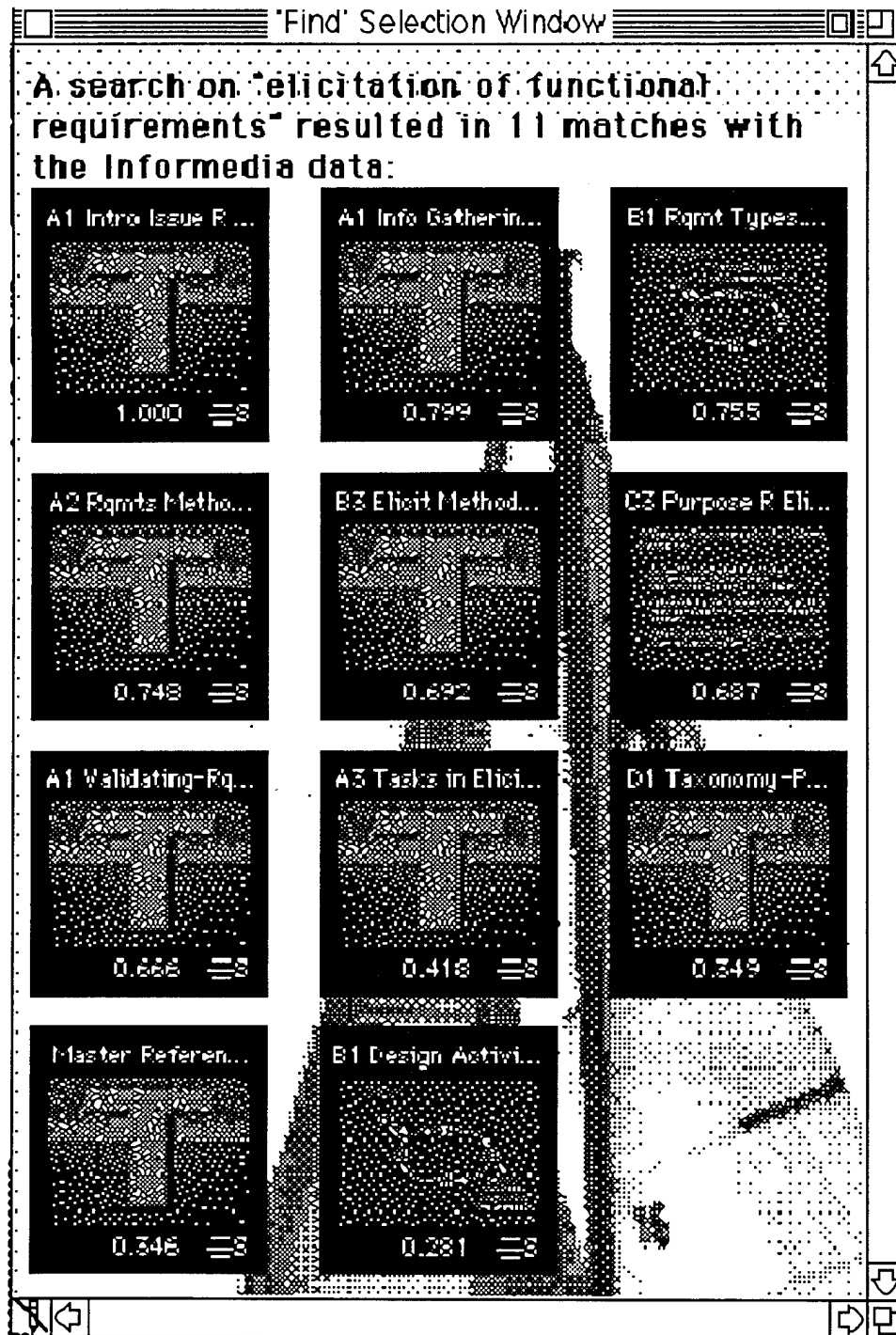


Figure 1: Example of a Find Selection Window

3.3 Easy Access to Find the Desired Information

The software engineer needs to be assisted in finding the information that he or she needs. There are a number of ways to find the desired information with SAIL on the Informedia DVLS. In the following discussion, the "clip" refers to the smallest piece of text, audio, graphic or video contained on the repository. They are hypertext-linked² together to create topics.

- Find - With "Find," the user can quickly ask for a subject and retrieve the information and access it in any media form without the clutter of other information. Reference information can be found easily with the "Find" command. The user can get into and out of the repository quickly.
- Synonym - It cannot be assumed that the software engineers will all use the same terminology. The Informedia DVLS search system uses synonyms to locate many possible matches.
- Buttons - The introductory graphics of SAIL on Informedia DVLS has "buttons" which users click to bring up the graphic, text, or video represented on the button. This enables the user to navigate quickly to the subject needed.
- Sequences - Text clips may contain a "See Also" section for navigation suggestions. Sequences of clips can be viewed in a pre-determined sequence.
- Next and Previous - Video clips contain "Next" and "Previous" buttons to help in navigating through the materials. Thus, sequences of video clips can be viewed in a pre-determined sequence. Users can always go back to clips that have been previously viewed.
- More Context - Video clips contain "More Context" buttons to allow the user to go to a higher level of detail. Some times this will take the user to the table of contents; other times, the user will be presented a complete video that has been skimmed. In all cases the user can return to the spot where the "More Context" button was pushed.

3.4 Produce a High Find/Seek Hit Ratio

In order for the software engineer to rely on the technology database rather than re-invent the technology, the search system has to quickly provide the information he or she needs. The user must have a satisfactory outcome the majority of the times that they use the system, otherwise they will continue to re-invent. He or she should either find the technology desired or one that can be adapted to be used.

This is the toughest requirement because volume of the information required in the library to satisfy it. The ratio of the number of times a satisfactory outcome is found (success) divided by the number of times the software engineer has gone to the repository to find information (attempts) is referred to as the "hit ratio" (see Figure 2).

². The hypertext link lets the viewer select the information (usually by clicking the mouse on a specific piece of text or a graphic). The selected information is then displayed on the screen.

If the hit ratio is too low the software engineer won't use the system. The exact level where the software engineer will lose interest has not been determined.

As the system is developed and the repository grows, there will be areas where the software engineer can expect a higher hit ratio than others. Informedia experiments with voice input have demonstrated that intelligent speech recognition will speed up the access to the desired information by providing the ability to quickly tailor, adapt, and expand the request until the user finds the desired information. It is easier for the software engineer to explore different avenues using voice access, thus increasing the hit ratio.

$$\text{Hit Ratio} = \frac{\text{Successes}}{\text{Attempts}}$$

Figure 2: Access Hit Ratio

3.5 Be Able to Expand With Project Information and Knowledge

The system needs to provide a way for people to both incorporate and access project information. Such presentations are frequently in multimedia form.

In today's world, it is up to the software engineer to take good notes on what is said and to keep track of these notes along with the presentation materials. Frequently, presentations are not given when the software engineer needs the information; therefore, he or she needs to properly interpret the notes to apply to his or her development effort. This process causes a lot of errors to enter the target system. The SAIL system provides a way to electronically record and access such multimedia information.

Project information needs to be available to the software engineer at his or her fingertips. Placing this type of information onto the repository is addressed in detail in a technical report on how to add technical information to the Informedia library [Hallman 96]. It addresses the archival strategies of documentation and recording of project information on a SAIL repository using the following five approaches:

- document a meeting with video
- document a meeting with audio
- document a slide presentation
- adding an existing slide presentation
- adding a text document quickly

3.6 Be Adaptable to Current Development Activities

Project information needs to be available to the software engineer at his or her fingertips in such a manner that it can adapt into the software engineer's current project. The more adaptable the information is, the more people will use it. Electronic information can be easily tailored when needed.

SAIL with Informedia DVLS provides the information in a form that can be copied or printed for to use in development activities, so users can identify the need, locate the materials, copy the information on the screen, paste the information into the work in progress, and revise the work or fill in relevant data as is needed to adapt it.

3.7 Be Organized Into Sub-libraries

The size of the potential repository can cause accessing problems. These can be solved by organizing the content into libraries and sub-libraries and placing them on large hard disk drives or on CDROM disks. The software engineer can then put the less used libraries and sub-libraries into his or her CDROM drive when needed.

Having divided the repository into libraries and sub-libraries, the support system needs to be able to inform the user when a desired subject is on another library. With SAIL, a directory called "Mass Media Indexes," containing a file for each disk with names of the files on the library or sub-library, is placed in the SAIL directory. Keywords for each sub-library can be added to the directory. While searching, the user can also determine whether the non-resident media should be accessed. These libraries or sub-libraries may be temporarily stored on large hard drives such as gigabyte drives or they can be permanently placed on CDROM disks.³

Appendix A on page 31 can be used as a guideline for dividing a repository into libraries and sub-libraries.

3.8 Provide for Both Learning and Reference

Information has to be organized so that it can be readily accessed for reference, but it also has to support the software engineer who is learning how to use the technology.

In SAIL, detailed descriptions of the subject matter refer to materials which help the software engineer to learn the technology; thus, the engineer gets used to seeing the signs for reference materials. If the engineer needs a refresher on how to use it, the reference materials have connections back to the descriptions. This is addressed further in Section 6 on page 21.

³. This is especially easy if an organization has CDROM writing equipment.

3.9 Shortcomings

Several things are missing from the SAIL with Informedia DVLS:

- Informedia is in its infancy. Most of its efforts have been aimed at solving the problems associated with finding information inside videos by verbal and visual means. Very little has been done outside of SAIL to collect the necessary software engineering information.
- Informedia has not tackled the problems of searching the Web for relevant information. It is oriented toward searching the known libraries, not unknown libraries, as would be necessary with searching the Web.
- Technical library information is not generally available in electronic form. The state-of-the-practice is documented in the technical periodicals and books normally available in physical libraries or from the publisher. However, more and more frequently, publishers are beginning to make their publications available on CDROMs.⁴

⁴ The proceedings of the Fifth Annual International Symposium of the National Council on Systems Engineering are published on an electronic CDROM as well as on paper.

4 Fingertip Access with the World Wide Web

The World Wide Web [Berners-Lee 94], through browsers similar to Mosaic [Schatz 94] and Netscape Navigator™, has the potential of providing fingertip access to the vast libraries of technical information. The Web libraries can be browsed until something is found that sounds similar to what is needed. The information can then be downloaded and viewed at the software engineer's work station.

4.1 The Web and Software Engineering

The Web is an advanced communication system available on the Internet. In layman's terms, it provides a mechanism to access a large number of Internet locations or "Web sites" throughout the world through hypertext links⁵ which eliminate the necessity of knowing where the site is located. Information available at these sites can be downloaded by simply clicking on the hypertext link. A large number of Web sites are now on Internet and the number is increasing. Organizations both private and public have made Web sites, which are available to the worldwide networking community. The information at these Web sites is in many multimedia formats and standards are evolving to maximize their use. Text and graphics are common forms of media on the web, but videos will also appear occasionally. Downloading time can be fast or slow depending on the type of media, its size, and its popularity.

Access to some, but not all, information needed by software engineers is available on the Web. But hardware and software companies have begun placing specification information about their products at their Web sites, which provides important information to software engineers—what hardware or software is available that can be used and built upon. Also, some non-profit organizations have placed their technical reports on the Web. In other cases, only a list of the contents of their records exist. The amount of information available on the Web is growing daily.

4.2 Accessing Information on the Web

You need several things to start accessing the Web (besides having access to Internet and the Web). First, you need to know what Web sites potentially contain the desired information. Second, you need to know related terminology needed to find titles that might contain the technical subject matter that is needed. Third you need patience as you surf the Web from library to library and browse each to find the technical article that might contain the information needed. And you will need to scan the entire articles to find the place where the exact technology exists. Occasionally, perhaps more often than not, you will not find the information you thought you had and you will have to return to the Web to continue to browse.

5. On the Web, hypertext linking allows information to be downloaded from a Web site. The viewer doesn't need to know where the document exists or even how to find it; the hypertext-link provides all of the information necessary to make the connection. The support system obtains the multimedia documents available through such links.

You need to know what you are looking for so you can surf the Web and browse for what you need.

4.3 Experience Makes it Better

With experience on the Web, you will be able to set up home pages, "hotlists," and bookmarks that will speed up your travels through the Web sites.

Over time, more and more technical information will be available. This is especially true regarding available hardware and COTS (commercial off the shelf).

4.4 Shortcomings

Several things are missing from the Web:

- Web browsers are unable to find the relevant reference materials. The user has to interactively guide the browsing until the needed information is located. Some Web sites provide the capability to search by keyword on the titles of the documents in the repository before downloading a specific document [Huff 94], but searching within a document is rarely available.
- The technical library information is not generally available in electronic form. The state-of-the-practice is documented in the technical periodicals and books normally available in physical libraries or from the publisher.

In addition, the time it takes to search for and download the desired technical information can interfere with the software engineer's work in progress on his or her desktop.

5 Fingertip Access with SAIL and the Web

A combination of the SAIL approach to Informedia DVLS (IDVLS) and the Web could result in a very powerful tool for the software engineer. A software development organization can use the Web to collect relevant information as it becomes available. They could then place it in their local repository and scan in other relevant hard copy documents into the Informedia DVLS Library, using the Informedia DVLS to search and find the needed information in real time. This can solve the fingertip access and learning problem.

5.1 At the Beginning of the Project

At the beginning of the project, it is useful to

- use the Web to find and download relevant articles and place them on the Informedia DVLS library. It may be possible to establish collaborations with organizations found on the Web.
- purchase the CDROMs from the relevant publishing houses and make them available to Informedia DVLS
- scan into the library those articles that are relevant but not available on the Web. Attention must be paid to the copyright. Appropriate fees may be required.
- collect technical information that is relevant to the new project from other projects and standards groups in the organization and place it on the Informedia DVLS library

5.2 During the Project

During of the project, it is useful to

- occasionally browse the Web for new materials provided; locate relevant information and download or acquire a copy of it and make it available to Informedia DVLS
- occasionally purchase new CDROMs from the relevant publishing houses and make them available to Informedia DVLS
- scan new articles that are relevant into the library. Again, adhere to copyright requirements

5.3 On a Daily Basis

Use the Informedia DVLS on a daily basis to satisfy software engineers' information and training needs. When information is missing that is known to exist, it can be found and added to the Informedia DVLS Library. This may be accomplished by the individual, the project or a service organization.

5.4 Fingertip Access in Large Organizations

With large organizations, an information service can be established to provide the repository updating service. Since many of the functions required to prepare a fingertip information and learning approach are similar to a librarian's duties, it may be appropriate to have a librarian perform this service for the entire organization. Also, more common techniques and standards could be cut on CDROMs and distributed to the individuals for use at their workstations, or a common server could be set up to contain the information for all of the software engineers in the organization.

As the Informedia DVLS repository gets too large, local CDROMs with specific areas of interest can be cut and distributed. This is a natural expansion of the services provided by the local library. They already know the procedure to make copies of copyrighted information. They also have access to technical sources outside of the Web.

6 The SAIL Approach to Using Informedia

The SAIL experiment, a "System for Access to information and Learning," undertaken in the Software Engineering Information Modeling Project (SEIM) at SEI, examined the technical feasibility of creating a system that will provide software engineers with access to technical information at their desktop. It uses the Informedia DVLS as the vehicle for storage, access, and manipulation of multimedia information about requirements elicitation.

The objectives of the SAIL experiment were stated as:

Develop a model of a support system that can provide access to a vast library of information in an understandable format, that is available when needed, and is efficient to use.

- The system shall be able to access, capture and preserve knowledge about the development of software intensive systems.
- The system shall be able to be used to learn useful skills that the software engineer needs to be an effective professional.
- The system shall be able to provide the software engineer (at all skill levels) with a complete range of technology about the development of software.

The experiment showed that the same technical information can serve two purposes: as an access to reference information and as a tool to learn how to use the technology. Because of the potentially large volume of the technical information addressed, as described in Section 2.3 on page 6, the experiment also examined how the information should be organized both within a specific subject or major topic and how major topics can be collected into volumes of information.

6.1 Access and Learning

Several approaches to accessing and learning about a technology are supported in SAIL:

1. A software engineer may need access to reference materials to use on his or her project.
2. A software engineer may be in the middle of defining some procedure and realize he or she doesn't remember enough about a particular algorithm or tool and needs some guidance to use it in this instance.
3. A software engineer may need training in a specific technology and has come to SAIL for that learning experience.
4. A software engineer may be interested in learning something about an unfamiliar subject and wants to understand the concept but doesn't need to use it.

These situations are elaborated upon below.

(1) When a software engineer needs access to reference materials, it may be to look something up in a table as in the specification for a hardware device, or to get copies of forms that are to be used on the project, or to get a logical description of an algorithm. For whatever reason, the software engineers need to access to the reference materials immediately. SAIL on Informedia DVLS provides several ways for them to find the desired information. If the engineer doesn't know exactly what the reference material is called, he or she can use the "Find" function to navigate to the right material.

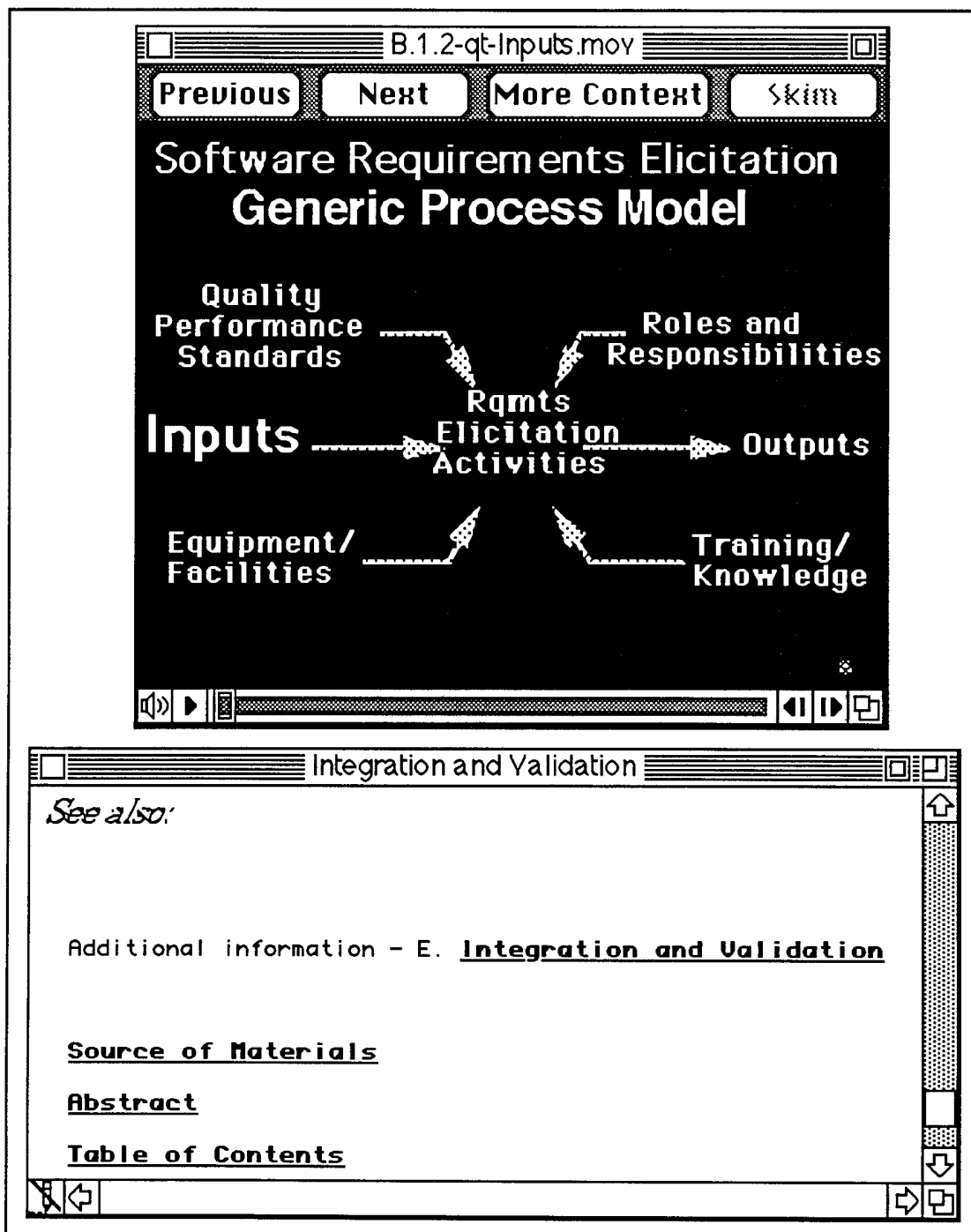


Figure 3: Accessing More Information About a Reference

The "Master Terminology List" may also be helpful in triggering the access. If the engineer knows what he or she is looking for, the "Master Reference" might be the fastest way to find it. If the engineer is looking for a reference that he or she has used frequently or is associated with one that is used often, the "SAIL Directory" or the graphic interface can be used to quickly access the outline of the major topic. The software engineer can also make his own "hotlist" to frequently used references.

Because the reference materials are selected from the detailed descriptions of the technology, the engineer can review the technical descriptions about the reference by using the "Previous" and "Next" buttons on the video window or the "See also" screen at the end of the text. (See Figure 3 for an example of these two approaches to navigation.)

(2) When the software engineer is in the middle of defining some procedure and needs some guidance to use a particular algorithm, he or she can use the "Find" function to locate and navigate to the desired description. All of the alternatives described in (1) above are also helpful.

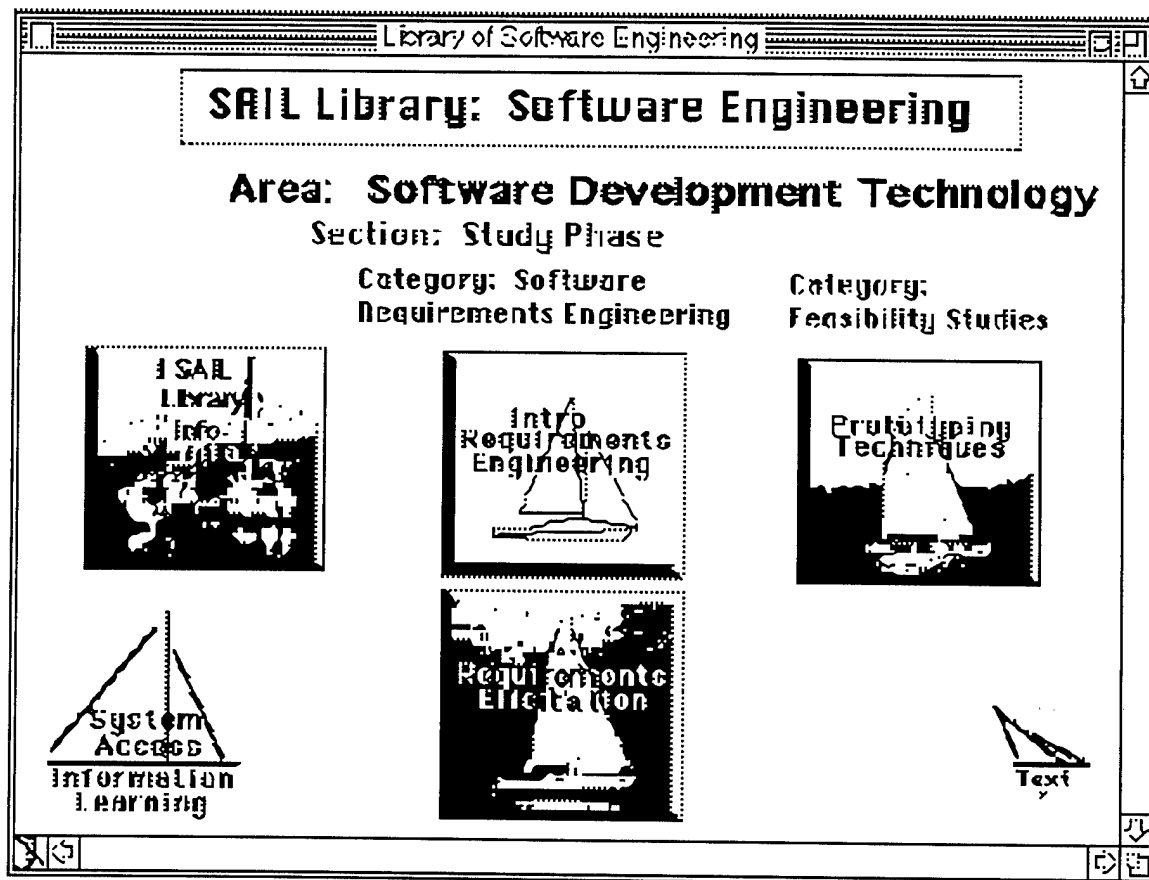


Figure 4: Library of Software Engineering Window

(3) When the software engineer is in need of training in a specific technology and knows what major topic contains that training, he or she can go to the "SAIL Directory" to find that training. If he or she doesn't know the specific major topic, the "SAIL Directory" is the right place to start. SAIL provides the directory in both text and hierarchical graphics format. They can access the graphic form from the opening Informedia window by clicking the "Library of Software Engineering" button. (Figure 4 is a typical graphic interface to the volumes that contain major topics on the library.) Clicking the buttons provides access to all of the major topics in the volume.

Access to the textual form of the "SAIL Directory" can be attained by clicking the "Text" button on the lower right of the graphic form of the directory.

(4) When the software engineer is only interested in learning about the concept of an unfamiliar subject, he or she can find related subjects using the "Find" function to locate a number of topics containing related information, and then browse through these until satisfied. By browsing the outlines containing the related information, he or she may be able to find the desired information or decide that it is unneeded. Browsing through the graphical directory may also help.

6.2 Library Organization

The basic unit of technical information is the "Major Topic." It may contain a technical report, an audio, or a video on a particular subject. A major topic represents one source of the technical subject and is usually a single article, technical report, audio, or video. It may also be a collection of graphics or reference materials. There may be separate major topics on the same subject, each perhaps having a specific point of view. All of the information needed to understand and use the technology is contained in the parts of the major topic. (Figure 5 summarizes the possible content of a major topic.)

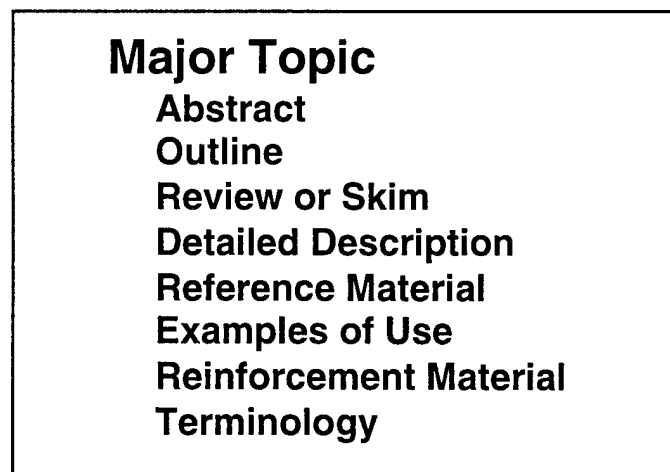


Figure 5: Parts of a Major Topic

Some examples of major topics from the SAIL experiment are

- Software Requirements Engineering at Texas Instruments⁶
- Issues in Requirements Elicitation [Christel 92a]
- Software Requirements Elicitation at Texas Instruments
- Group Development Methods [Hill 92]
- A Classification and Bibliography of Prototyping [Wood 92]

The major topic is often broken down into several "Themes." Each theme addresses the major topic from a specific direction. For example

Major Topic: Software Requirements Elicitation at Texas Instruments

Themes: Intro to Requirements Elicitation Process

Themes: Intro to Requirements Elicitation Methods

Themes: Reference materials

The lowest unit of material is referred to as the "Topic." It may use several media elements in order to present its intent. A theme may require several topics to cover the intended technical description. For example

Themes: Intro to Requirements Elicitation Methods

Topic: Questionnaires

Topic: Interviews

Topic: Operational Scenarios

Topic: Prototypes and Models

Topic: Brainstorming

Topic: Joint Application Design (JAD)

Topic: Quality Function Deployment (QFD)

Topic: Extraction from Documents

Topic: Observations

⁶ This is based on the video tape presentation, "Introduction to Software Requirements Elicitation," created at the Software Engineering Institute, Carnegie Mellon University, in conjunction with Texas Instruments, Inc., 1993.

More intensive presentations of these topics being introduced in this theme may be contained in a major topic or theme dedicated to that method. For example, in a fully populated library, there would be major topics on most of these methods describing in detail how to use the method. The information is structured hierarchically and linked together with hypertext links, giving the user control of what he or she wants to see, hear, or read.

The Major Topics are grouped together in an organized manner and reside in the appropriate part of the library. There may be many documents or presentations on a specific subject. These major topics are grouped into a "Volume" in order to facilitate more efficient access to similar topics. There often are several approaches to a describing a subject. For example, a volume on requirements elicitation might contain descriptions of several alternative approaches. One might recommend individual interviews with the various parties that needed to be contacted, while another might recommend a massive session with all appropriate parties to solicit all requirements at one time. A major topic would be used to indicate which alternative was being discussed. For example:

Volume: Software Requirements Elicitation

Major Topic: Issues in Requirements Elicitation

Major Topic: Software Requirements Elicitation at Texas Instruments

Major Topic: Group Development Methods

Higher-level groupings are desirable as more and more information is available in the repository. Grouping may be for logical reasons, to keep source types separate, or to facilitate distribution. Appendix A on page 31 uses a logical grouping of Area, Section and Category as levels of above the Volume. This approach gives the hierarchy of groupings shown in Figure 6.

Other approaches to groupings may use a different hierarchy. For example: phase of development, type of software developed, historical significance, source of materials, level of confidence in materials, etc.

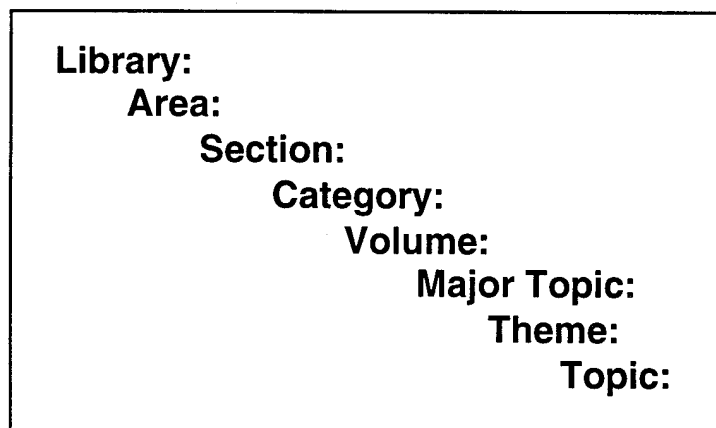


Figure 6: Logical Hierarchy of the Repository

6.3 Facilitating Easy Access

There are many approaches for finding and accessing information. This list in alphabetical order shows some of the facilities provided. Items marked with (I) are provided by the Informedia DVLS and therefore are available to all users of Informedia. Items marked with an (S) are provided by the subject matter person using the SAIL approach to creating the library. Items marked (I and S) are provided by the person using the SAIL approach and are supported by a facility in the Informedia DVLS system:

- Buttons on Graphics (I and S)
- "Find" method of searching (I)
- "Go Back" pull down list (I)
- Hypertext-links embedded in text clips (I and S)
- "Mass Media Indexes" (S)
- "Master Reference List" (S)
- "Master Terminology List" (S)
- "Previous" "Next" and "More Context" buttons on video clips (I)
- Outlines in Major Topics and Themes (S)
- SAIL Directory (graphic and text) (S)
- SAIL Table of Contents (S)
- "See also:" sections in text clips (S)
- Skims (I)
- Synonyms (I)
- Tutorial on "How to use Informedia" (S)
- Tutorial on "How to use SAIL" (S)

All of these have been discussed earlier in this report. The how and where are addressed in [Hallman 96] on creating multimedia technical libraries.

7 Summary

It is expected that the software engineer will be more effective when the technologies he or she uses are available at his or her finger tips. SAIL on Informedia DVLS has demonstrated that fingertip access is feasible and with it, fingertip learning can be provided to the software engineer. The populating of the repository necessary to satisfy the needs of the software engineer is incomplete. Using the World Wide Web to access and obtain some of the materials can now be accomplished. This is especially true for hardware and software specification information. The informedia Digital Video Library System and the SAIL approach to creating the technical library provide an effective fingertip access and learning system. Project information can be added to the repository along with company developed technical know-how. Fingertip access to the knowledge of the profession at the fingertips of the software engineer will greatly improve software engineers' productivity and the quality of the products they are developing.

Appendix A SE-SAIL Library Outline: An Example

The following is an incomplete example of what the SAIL library for software engineering might contain. It is provided to give a better understanding of the information and training needs of the software engineer.

SAIL

Library: Software Engineering

Area: Engineering Technology

Area: Software Development Technology

Section: Study Phase

Category: Business Case Development

Category: Software Requirements Engineering

Volume: Introduction to Software Requirements Engineering

Volume: Software Requirements Elicitation

Volume: Software Requirements Analysis

Volume: Software Requirements Specification

Volume: Validation of Software Requirements

Volume: Software Requirements Engineering Products

Volume: Software Requirements Tools

Category: Feasibility Analysis

Volume: Concept Development and Analysis

Volume: Technical Analysis

Volume: Schedule Analysis

Volume: Cost Analysis

Volume: Trade-off analysis

Section: Design Phase

Category: System Concept Definition

Volume: Objectives Development

Volume: Environment Definition

Volume: Operations Concept Development

Volume: Constraint Definition

Volume: Testing Objectives

Volume: Engineering Process Definition

Category: System Architecture Development

Volume: Conceptual Architecture Development

Volume: Functional Analysis and Decomposition

Volume: Boundaries and Interfaces

Volume: Architectural Models

Volume: Requirements Allocation and Tracking

Volume: Architectural Standards

Volume: Testing Architecture Development

Category: System Design

Volume: Design Methodologies

Volume: Design Documentation

Volume: Design Standards and Rules

Volume: Testing System Design

Category: Tools

Section: Implementation Phase

Category: Implementation Technologies

Category: Validation Technologies

Category: Verification Technologies

Section: Post-Deployment Phase

Category: Transition

Category: Metrics

Category: Error Tracking

Category: Reverse Engineering

Area: Software Process Technology

Section: Life-Cycles

Section: Process Control Technologies

Section: Maturity Models

Category: Capability Maturity Models

Volume: CMM (SEI)

Volume: SE-CMM (SEI)

Category: Other Models

Area: Software Management Technology

Library: Computer Science

Area: Languages

Area: Algorithms

Section: Analysis of Algorithms

Section: Proof of Correctness

Area: Data

Section: Data Structures

Section: Data Storage - Data Bases

Section: Data Manipulation

Category: Sorting and Merging

Category: Searching

Area: Systems Architectures

Section: Data Flow Architectures

Category: Batch Sequential

Category: Pipes and Filters, etc.

Section: Communicating Process Architectures

Category: Message Passing

Category: Client-Server Architectures

Category: Communicating Sequential Processes

Section: Call-and-Return Architectures

Category: Event-based, Implicit Invocation

Category: Main Program and Subroutines

Category: Data Abstraction and Information Hiding

Category: Object-Oriented Design

Category: Layered Architectures: Network Protocols

Section: Virtual Machines

Category: Table Driven Interpreters

Category: Rule-based systems

Section: Data-Centered Systems (Repositories)

Category: Transactional Databases, Blackboards

Section: Heterogeneous Architectures

Section: Distributed Systems Architectures

Category: Characteristics of Distributed Systems

Category: Parallel or Concurrent Programs

Category: Networked Computing

Volume: Networks

Volume: Communications Protocols

Category: Cooperative Computing

Section: Concurrent Systems Architectures

Section: Mixed Use of Idioms in Software Architectures

Area: Operating Systems

Area: Storage Management Algorithms

Library: Software

Area: Software Descriptions

Area: Reusable Software

Section: Reusable Designs

Section: Reusable Code

Section: Reusable Tests

Area: Domain Specific Architectures

Section: Finance

Section: Avionics

Section: Guidance

Section: Robotics

Section: Company Management

Section: Manufacturing

etc.

Area: Software Specifications

Section: Applications

Section: Systems

Section: Tools

Library: Hardware - Computer Related

Area: Architectures

Area: Hardware Specifications

Library: Mathematics

Area: Algebra

Area: Discrete Math

Area: Set Theory

Area: Probability

Area: Statistics

Area: Formal Methods

Library: Local Information

Area: Standards

Section: Software Process Technology

Section: Study Phase

Section: Design Phase

Section: Implementation Phase

Section: Post-Deployment Phase

Library: Project XYZ Information

Area: Raw Information

Section: Requirements Info

Section: Design Info

Section: Implementation Info

Section: Post-Deployment Info

Area: Project Approved Information

Section: Requirements Info

Section: Design Info

Section: Implementation Info

Section: Post-Deployment Info

Area: Externally Approved Information

Section: Requirements Info

Section: Design Info

Section: Implementation Info

Section: Post-Deployment Info

Appendix B Content of a Major Topic

All of the information needed to understand and use the technology is contained in the parts of the major topic. These may be distributed across the themes and topics of the major topic but all are part of the major topic. The types of information needed by the users are

- abstract
- outline
- review or skim
- detailed descriptions
- reference material
- examples of use
- reinforcement materials
- terminology

The subject matter expert who creates or prepares the materials for the library has various approaches to do this. He or she may start with a group of reference materials and then to create the supporting materials in a bottom up approach. He or she may start with an outline and work down the outline creating the materials in a top down approach. Occasionally the materials will have already been recorded. [Hallman 96] describes the various approaches to placing information on the repository.

B.1 Abstract Description

Within each major topic is an abstract describing the technical material needed to conceptually understand the technology. It introduces the major topic giving an abstract view of the material in terms a novice in the subject materials can understand. It contains hypertext-links to the outline of the major topic. It identifies the author/speaker and provides the appropriate credits for the materials abstracted.

B.2 Outline

The outline of the material in the major topic is presented to the practitioner from a technological point of view. The outline looks more like an hierarchy than a nested list. It is more like a work breakdown structure giving the user options rather than having to traverse all of the materials to get to the desired part.

Hypertext-links are provided using HTML [Deuel 93] to give immediate access to the material where each outline entity is covered.

B.3 Detailed Description

Within each major topic, there are detailed descriptions of the technical material. These are presented in a manner that will prepare the user to utilize and/or create extensions to the topic. It is usually expressed in the terminology of a novice on that technology.

When the detailed description is being created specifically for fingertip learning, the following guidelines are useful.

- Quickly introduce the subject and the organization of the material in this description. The overriding guidelines are efficiency and effectiveness.
- Describe the technology in terms that a novice can understand. Keep it simple and concise. Use reinforcement information and examples to assist the novice in understanding. When structured and linked by hypertext,⁷ the system will bring the proper presentation together for the novice or the practitioner.
- Use multimedia to get the most effective retention.⁸
- Point users to related subjects that they may want to investigate. These do not have to be lined with hypertext since the Informedia system allows the user to do a search on any subject at any time and quickly brings up the requested subject. By just providing a pointer to trigger users' interest, the user has the option to look at the suggested materials or ignore them.
- Do not duplicate the information in the reference parts of the topic; instead, pull users in with hypertext links. It may be necessary to provide additional information for the user to understand and learn how to use the reference material.
- Include materials that enable the novice user to learn how to be a practitioner. Make viewing this material optional. Make use of the materials in the reinforcement part of the major topic.
- Examples and small case studies may be necessary to provide the novice with understanding and insight. Have these available through hypertext links. The user can bring them in at the appropriate time.

[Hallman 96] provides detailed approaches to the addition of existing and new multimedia materials to the repository.

⁷. These hypertext-links may be in the form of HTML links in text, buttons on graphics, or "Next/Previous" buttons on videos and audios.

⁸. Interactive animation's primary effect is the reduction of the cognitive load by exploiting the human perceptual system. It helps to provide insights into the relationships between substructures. This has been seen in CAD/-CAM systems also [Robertson 91].

B.4 Review or Skim

Each major topic may contain a review or a skim of the technical material covered in the detailed description part. This is intended to give the highlights of the materials covered so that the user can get a quick review of the material. It is expressed in the terminology of the full description. This provides an opportunity for the novice to see if he or she understands the terminology necessary to practice the technology.

With audio or videos, a skim of the media is helpful. A skim is a selection of very short clips from the original tape that gives the potential user some insight into the content of the audio or video. Thus, in several minutes the long video can be reviewed. The Informedia project at Carnegie Mellon University has experimented with video skims, which can be created manually, and also with techniques for creating skims automatically.

B.5 Reference Material

Within each major topic, there are usually a number of reference materials. This includes all of the tables and graphs necessary to use the technology. It needs to be organized in such a manner that a person who has an understanding of the material can use it without looking at the other parts of the topic. After the detailed descriptions are completed, the reference materials are identified and summarized in a table type format (usually as part of the outline) providing hypertext links to the single copy of the materials. Several entries may point to the same reference materials. This provides more than one way for the user to find the needed reference.

The name of the references, along with hypertext links to the reference and the abstract of the major topic containing them, are placed alphabetically in the "Master Reference" of the SAIL Library. (Figure B-1 shows an example of the entries in the Master Reference.) Note that the hypertext links appear dark and are underlined. Also note that the link itself identifies the media of the reference. In some cases, the reference will appear in multiple media forms. The last hypertext link for each entry is to the abstract of the major topic containing the reference. By using the name of the author and the year of creation, as in a bibliography reference, the user is given the ability to get additional insight immediately concerning the reference. By clicking on the link, a description of the major topic is provided.

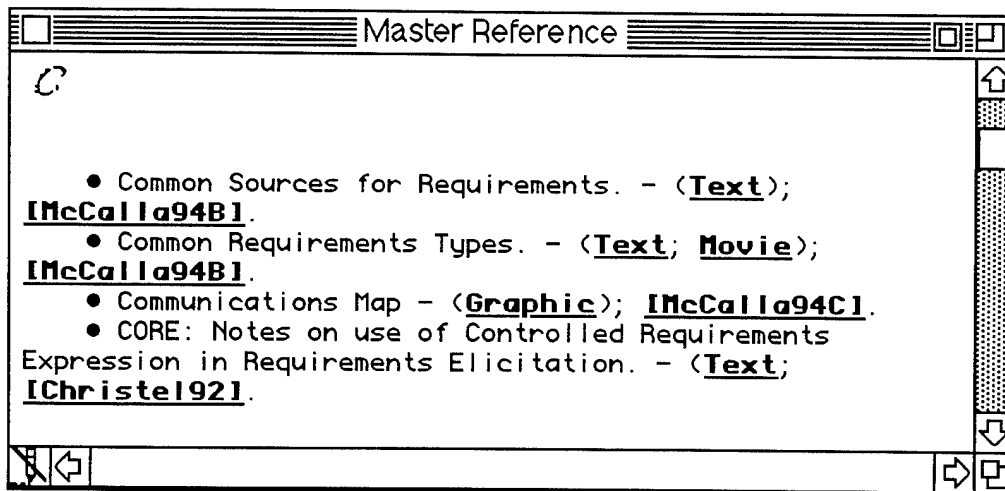


Figure B-1: Example of the Master Reference List

B.6 Examples of Use

Within each major topic, there may be examples of use for the topic or reference material. This allows the novice to learn by example and get a better understanding of the methodology. It provides examples of the usage of the topic or reference material. This may be as simple as a copy of a filled out form that appears in the reference materials or as complicated as an example of the complete use of the topic. Thus, it will help prepare the user to use and/or create extensions to the topic.

B.7 Reinforcement Material

Within each major topic, there may be reinforcement materials for the technology that will help the novice remember the materials presented in the topic. These are usually kept separate from the other materials. It allows a person using the reference materials who needs the some specific detailed information to avoid being bogged down with reinforcement materials unless he or she wants or needs them.

B.8 Terminology

Within each major topic, there may be a topic containing the definitions of terms introduced in the major topic. Those terms that are unique to the topic being discussed are described in a short one or two sentence explanation. The terms are also placed alphabetically, along with a hypertext-link to where a more in-depth knowledge of the use of the term may be found, into the SAIL "Master Terminology List." Occasionally, the description of the term is extensive. Then, only the hypertext link to the definition will appear.

Figure B-2 shows an example of the entries in the Master Terminology List. Note the hypertext links. Note also that terms may have slightly different definitions depending on the technical materials using the term.

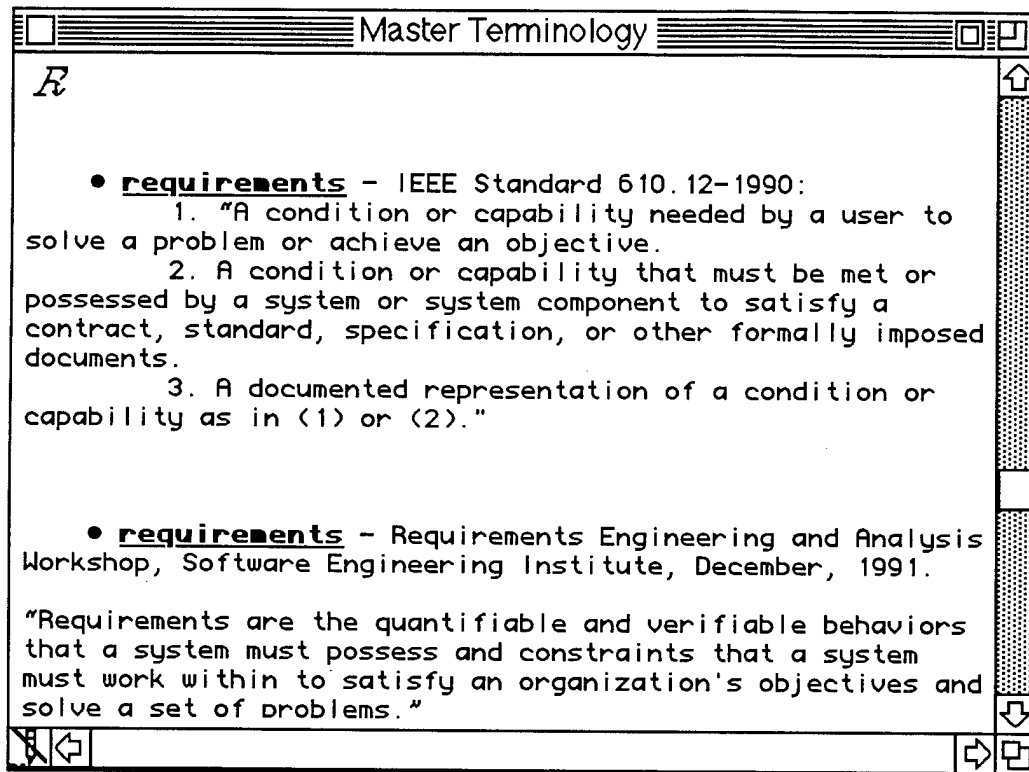


Figure B-2: Example of the Master Terminology List

References

- [Berners-Lee 94] Berners-Lee, T.; Calliau, R.; Loutonen, A.; Frystyk, N.; & Secret, A. "The World Wide Web." *Communications of the ACM* 37, 8 (August 1994): 76-82.
- [Christel 91] Christel, M. *A Comparative Evaluation of Digital Video Interactive Interfaces in the Delivery of a Code Inspection Course*, Ph.D. Thesis, Georgia Institute of Technology, Atlanta, Ga., 1991.
- [Christel 92a] Christel, M.G. & Kang, K.C. *Issues in Requirements Elicitation*. (CMU/SEI-92-TR-12, ADA258932). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1992.
- [Christel 92b] Christel, M. & Stevens, S. "Rule Base and Digital Video Technologies Applied to Training Simulations," *Software Engineering Institute Technical Review '92* (CMU/SEI-92-REV). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1992.
- [Christel 93] Christel, M. G.; Wood, D. P.; & Stevens, S. M. *AMORE: The Advanced Multimedia Organizer for Requirements Elicitation* (CMU/SEI-93-TR-12, ADA268058). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1993.
- [Christel 95] Christel, M.; Kanade, T.; Mauldin, M.; Reddy, R.; Sirbu, M.; Stevens, S.; & Wactlar, H. "Informedia Digital Video Library," *Communications of the ACM* 38, 4 (April 1995): 57-58.
- [Deuel 93] Deuel, P. M. *HTML Reference Guide*. Potsdam, NY: Clarkson University, 1993.
- [Hallman 96] Hallman, H.K. *Multimedia Technical Libraries: Informedia Digital Video Library System* (CMU/SEI-95-TR-006). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1996.
- [Hill 92] Hill, I.M. "Group Development Methods," Lecture 8, *Software Requirements Engineering* (CE-SRE-01-08). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1992. VideoTape 33:16 minutes.
- [Huff 94] Huff, C. C. *Spinning a Web: Publishing the SEI Software Configuration Management Research on the World Wide Web* (CMU/SEI-94-TR-18, ADA288708). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1994.

- [Perry 84] Perry, Robert H., ed. *Perry's Chemical Engineers' Handbook*, sixth edition. New York: McGraw-Hill, 1984.
- [Robertson 91] Robertson, G. G.; Mackinlay, J. D.; & Card, S. K. "Cone Trees: Animated 3D Visualizations of Hierarchical Information," 189-194. *CHI'91 Conference Proceedings*. New Orleans, La., April 27-May 2, 1991. New York: ACM Press 1991.
- [Schatz 94] Schatz, B.R. & Hardin, J.B. "NCSA Mosaic and the World Wide Web: Global Hypermedia Protocols for the Internet." *Science* 265, 5174 (August 12, 1994): 895-901.
- [Stevens 89] Stevens, S. "Intelligent Interactive Video Simulation of a Code Inspection." *Communications of the ACM* 32, 4 (July 1989): 832-843.
- [Stevens 92] Stevens, S. "Next Generation Network and Operating System Requirements for Continuous Time Media," *Network and Operating System Support for Digital Audio and Video*, Ralph Herrtwich, ed. New York: Springer-Verlag, Inc., 1992.
- [Stevens 95] Stevens, S.; Christel, M.; & Wactlar, H. "Informedia: Improving Access to Digital Video." *interactions* 1, 4 (October 1994): 67-71.
- [Studwell 90] Studwell, W. E. *Library of Congress Subject Headings: Philosophy, Practice and Perspectives*. New York: The Hatworth Press, 1990.
- [Wood 92] Wood, D. P. & Kang, K. C. *A Classification and Bibliography of Prototyping*, (CMU/SEI-92-TR-13, ADA258466). Pittsburgh, Pa: Software Engineering Institute, Carnegie Mellon University, 1992.
- [Wood 94] Wood, D. P.; Christel, M. G.; & Stevens, S. M. "A Multimedia Approach to Requirements Capture and Modeling," 53-56. *Proceedings of the International Conference on Requirements Engineering*. Colorado Springs, Co., April 18-22, 1994. Los Alamitos, Ca.: IEEE Computer Society Press, 1995.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS None		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-95-TR-018			5. MONITORING ORGANIZATION REPORT NUMBER(S) ESC-TR-95-018		
6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute		6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office		
6c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213			7b. ADDRESS (city, state, and zip code) HQ ESC/ENS 5 Eglin Street Hanscom AFB, MA 01731-2116		
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office		8b. OFFICE SYMBOL (if applicable) ESC/ENS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-95-C-0003		
8c. ADDRESS (city, state, and zip code)) Carnegie Mellon University Pittsburgh PA 15213			10. SOURCE OF FUNDING NOS.		
			PROGRAM ELEMENT NO 63756E	PROJECT NO. N/A	TASK NO N/A
			WORK UNIT NO. N/A		
11. TITLE (Include Security Classification) Fingertip Access to Software Engineering Information and Learning: SAIL on the Informedia DVLS					
12. PERSONAL AUTHOR(S) Harvey K. Hallman					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM TO		14. DATE OF REPORT (year, month, day) May 1996	
15. PAGE COUNT 44					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (continue on reverse of necessary and identify by block number) informedia multimedia software engineering libraries		
FIELD	GROUP	SUB. GR.			
19. ABSTRACT (continue on reverse if necessary and identify by block number) Practicing software engineers have difficulty accessing state-of-the-practice technologies in a timely manner. As a result, software engineers frequently re-invent the technology. Fingertip access to large amounts of information (project, state-of-the-practice, state-of-the-art, domain specific, etc.) should be provided so that software engineers perform their profession more effectively. The System for Access to Information and Learning (SAIL) approach on the Informedia™ Digital Video Library System (DVLS) demonstrates the feasibility of providing fingertip access to information and learning materials, using requirements elicitation as the technology base. Also, Informedia DVLS and the World Wide Web can supplement each other; by using the Web to gain access to certain areas of software engineering, we can begin to assemble libraries of technical information. Informedia DVLS can provide timely access to such information and can also be used to house current project materials. <div style="text-align: right;">(please turn over)</div>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS <input checked="" type="checkbox"/>			21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution		
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas R. Miller, Lt Col, USAF			22b. TELEPHONE NUMBER (include area code) (412) 268-7631		22c. OFFICE SYMBOL ESC/ENS (SEI)